The On-Chip 3-MB Subarray-Based Third-Level Cache on an Itanium Microprocessor

Don Weiss, John J. Wuu, and Victor Chin

***Abstract*—The 3-MB on-chip level three cache in the Itanium 2 Processor, built on an 0.18-μm, six-layer Al metal process, employs a subarray design style that efficiently utilizes available area and flexibly adapts to floor plan changes. Through a distributed decoding scheme and compact circuit design and layout, 85% array efficiency was achieved for the subarrays. In addition, various test and reliability features were included. The cache allows for a store and a load every four core cycles and has been characterized to operate above 1.2 GHz at 1.5 V and 110 °C. When running at 1.0 GHz, the cache provides a total bandwidth of 64 GB/s.**

***Index Terms*—Cache memories, memory, memory architectures, microprocessors, random-access memories.**

I. Introduction

FOR a given die size, floor planning and array efficiency have traditionally limited the size of caches on microprocessors. The Itanium 2 processor, built on a 0.18-μm, six-layer Al metal process, contains three levels of on-chip cache totaling more than 3.3 MB [1]. The on-chip third-level cache is a change from the first Itanium Processor Family processor's level three cache, which contains 4 MB and resides off-chip [2]. Moving the third-level cache onto the die offers many advantages. For example, it significantly reduces latency, eliminates the need for a back side bus, and reduces cost and complexity. However, these benefits come at the expense of size. In order to minimize this tradeoff, while the first-level cache on this processor is optimized for latency [3] and the second-level cache is optimized for bandwidth [4], this third-level cache is optimized for size. For a given die size, in order to achieve this goal, available area must be efficiently utilized. This 3-MB third-level cache employs a subarray design style that more easily utilizes available area, allowing for irregular block boundaries. It also provides flexibility for floor plan changes, which are inevitable over the design cycle of a processor chip. In addition, special care in circuit design and layout was taken to achieve 85% array efficiency for the subarrays. As a result, 3 MB of level three cache was incorporated onto the die. This single-ported, 12-way set-associative cache has a line size of 1024 b and can support 64 GB/s of maximum bandwidth.

II. Subarray Design Style

Most on-chip SRAMs are arranged into one or more large rectangular blocks. However, the major drawback to this traditional method is that it does not allow space to be efficiently utilized when the core is not regularly shaped. In other words, trying to fit a rectangular cache block next to an irregularly shaped core is similar to trying to fit a square peg into a round hole. As can be seen from the die photo of this processor (Fig. 1), the core is certainly not regularly shaped. Therefore, the subarray design style was chosen, which divided the 3-MB cache into 135 identical small subarrays. One hundred twenty-eight subarrays are used for data, five are used for ECC, and two are used for redundancy. The subarrays are tiled together to conform to the shape of the core and allow the front side bus I/O units, which contain drivers and receivers that interface with the system bus, to be distributed among the subarrays in four vertical stripes, located in between and next to the three lower subarray regions. The subarrays were able to efficiently fill up roughly 175 mm^2 of available space left by the core and the I/O units.

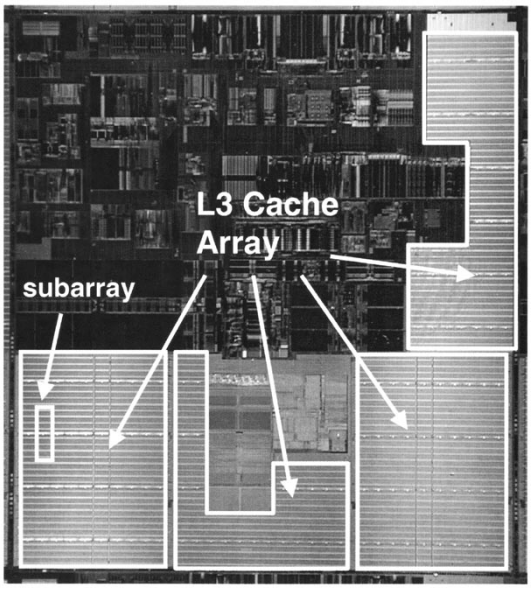


Fig. 1. Die photo.

Fig. 2 illustrates how the L3 subarrays are arranged logically. In a read operation, each subarray is responsible for producing 8 b, so the 128 data subarrays combine to provide the 1024-b line. The 8 b, while still inside the subarray, go through a 4 : 1 mux. Thus, the full line is returned to the core in four consecutive cycles, with the critical chunk returned first. The 4 : 1 mux is necessary to alleviate the wiring congestion between the bus cluster unit and the level two cache, and it also eases the wiring task within the L3 cache. The cache employs a shift redundancy

Manuscript received March 25, 2002; revised May 27, 2002.

D. Weiss and J. J. Wuu are with the Hewlett-Packard Company, Fort Collins, CO 80528 USA (e-mail: jjw@fc.hp.com).

V. Chin is with Intel Corporation, Santa Clara, CA 95052 USA.

Digital Object Identifier 10.1109/JSSC.2002.802354

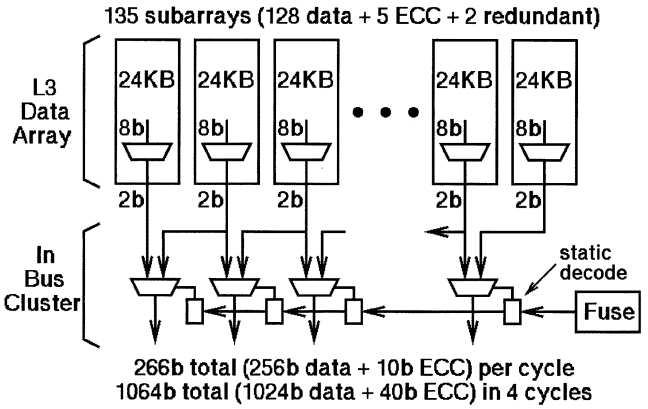


Fig. 2. Subarrays logical arrangement.

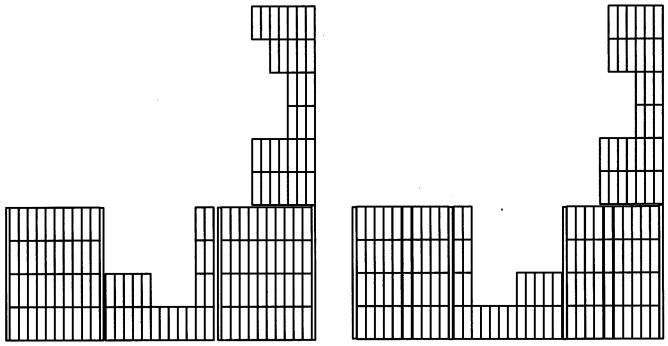


Fig. 3. Sample floor plans.

scheme [5]. Once the data returns to the bus cluster unit, it goes through the redundancy muxing, which is programmed through electrically programmable fuses. Not clearly shown in the figure is that there are two redundant subarrays, each responsible for repairing half of the normal subarrays. Data from a failing subarray is completely mapped out. As an example, if the second subarray from the left in Fig. 2 were defective, then data from the first subarray will simply flow through its mux, while the data from the third subarray will be rerouted through the second subarray's mux. Having an entire defective subarray mapped out allows us to not only repair bit cell, row, and column defects, it allows us to repair decoder and control logic failures as well. Also, since entire subarrays are mapped out, each individual subarray does not have to pay the penalties usually associated with having redundant columns and rows.

Another draw back to the traditional large rectangular cache design is that it does not respond well to floor plan changes, which are inevitable during the life of a microprocessor design cycle. If any part of the core grows, usually the cache has no choice but to be pushed out, thus increasing the die size. Therefore, another advantage of the subarray design style is that it can flexibly adapt to floor plan changes. Fig. 3 illustrates two different floor plans. The floor plan on the left is a "snap shot" taken about nine months before tape release; the floor plan on the right is the final floor plan. As can be seen, the bus cluster unit, occupying the inverted L-shaped hole among the lower subarrays, needed to flip about the y axis. Also, the floating point unit, located at the top of the chip, grew slightly larger than expected. In both cases, The L3 cache was able to accommodate the change by rearranging a few subarrays and did not impact the overall schedule.

III. Array Efficiency

Even with the subarrays efficiently filling up the available area, however, this alone is not enough to achieve the 3-MB goal for the L3 cache on this processor. Most high-density SRAMs today achieve an array density between 65% to 72%. With an array efficiency of 72%, 175 mm^2 of space would yield only about 2.7 MB of cache. So array efficiency is another knob that needed to be tweaked. The first thing to do for achieving high array density is to ensure compact cells and layouts, and significant efforts were made there. As in most high-speed, high-density SRAM designs, all the cells used in the design are completely custom, many of which will be shown in later parts of this paper. On top of having custom cells, the circuit and layout designers worked closely together to produce compact layouts. Feedback from layout is often used to retune, or sometimes completely redesign, circuits, and most circuits used in the L3 cache went through several such iterations.

Also, a large component to SRAM area overhead is usually associated with decoders and drivers, which typical SRAMs cannot go without. As designers are well aware, the RC load has a nonlinear relationship to driver size. As wires increase in length, drivers need to grow nonlinearly in order to keep the delay and edge rates acceptable. So, in this design, wires are partitioned with distributed decoding to only have small, manageable pieces of RC that can be driven by very small drivers. As a result, the decoders and drivers occupy only about 3% of the subarray area. Also, to reduce the need for having global repeater channels, which can be costly in terms of area, about 2.5% of the subarray area is allocated for global repeaters. So, with the distributed decoder scheme that allowed for small drivers to be used, and by having very compact circuits and layouts, an array efficiency of 85% was achieved for the subarray. The array efficiency number is calculated by dividing the area in a subarray occupied by RAM cells by the subarray's total area. Since each subarray does not contain any redundant elements, the efficiency number does not factor in the redundant subarrays. If one prefers to count redundancy elements as overhead, then the efficiency number for an average subarray would be around 84%. Outside of the subarrays, global repeaters result in additional inefficiency for the overall cache. However, the additional inefficiency due to the repeaters is only 0.17%, because most of the global repeaters are contained within the subarray, as discussed above.

IV. Subarray Architecture and Layout

Fig. 4 illustrates the distributed decoding scheme. The subarray is divided into eight groups, and each group is activated by its own "group clock." The group clocks are decoded and generated in the middle of each subarray. Having individual group clocks is advantageous because the group clocks are responsible for triggering all local signals, such as bitline precharge, local wordlines, and local column selects. Therefore, timing becomes a very localized problem. There are 96 global wordlines

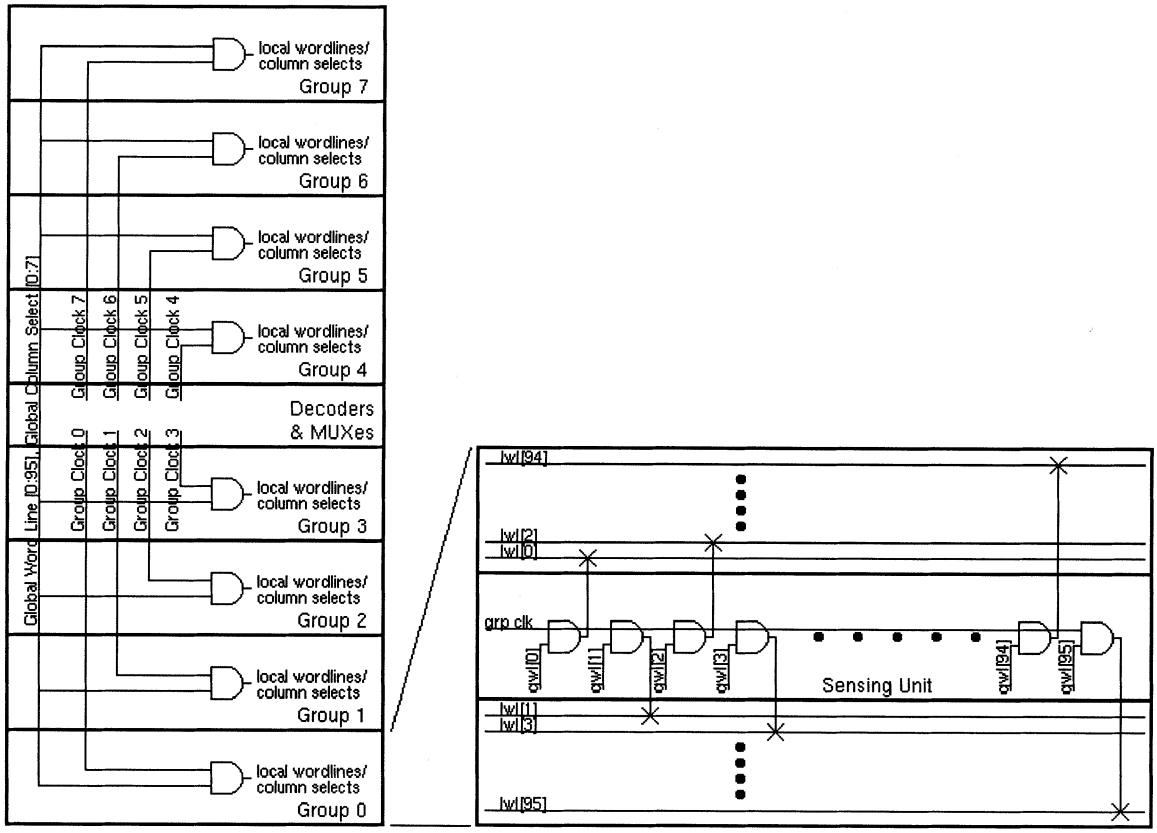


Fig. 4. Distributed decoding scheme.

decoded and generated in the middle of the subarrays, and those are ANDed with the group clocks in each group to generate the local wordlines. The 96 local wordlines are split on the two sides of the sensing unit, which is also where the small second-level decoders are hidden. Similarly, there are eight global column select signals generated in the middle of the subarrays, and ANDed with group clocks in the sensing units to generate the local column select signals. In all, there are 96 rows and 256 columns of six-transistor SRAM cells per group.

Fig. 5 contains the microphotograph of a subarray and shows how various components are laid out in the subarray. As mentioned before, the subarray is divided into eight groups, each with its sensing unit in the middle. In the middle of the subarray, one finds the first-level decoders that decode the index and way information sent from the tag unit into group clocks, global word lines, and global column selects. Underneath the first-level decoders are the write assembly blocks, which will be discussed later in this paper, followed by some muxes and latches used by read operations. Finally, there are some miscellaneous control logics. The miscellaneous control logics could have been spread out, allowing the subarray to be further compacted vertically. However, the decision was made to keep them bunched up for more efficient timing and leave the remaining area for global repeater space.

V. Circuit Details

The first-level decoders, shown in Fig. 6, take the index and way bits sent from the tag unit and generate the group clocks, global word lines, and column selects. The circuit needs to decode the index and way bits, latch the decoder output for a cycle, and then reset on the next cycle. The first part of this circuit is the static decoder, while the second part of this circuit is used throughout the chip as a static-to-dynamic circuit converter. The difference is that a half-frequency clock, which can be activated on any cycle, is used here. During the low phase of the half-frequency clock, the output stays low as the dynamic node is precharged high. The dynamic node evaluates based on the input in the short window between the clock rising and until the inverted clock disables the pull-down chain. A full feedback is required to hold the evaluated value for the remainder of the half-frequency clock phase. This circuit was chosen over the traditional master–slave flip-flops because it is faster and much smaller. It has very low hold time requirements, which is convenient because many subarrays reside near the source of the index and way routes, which could present race problems. The second level decoders are small static AND gates, and, as mentioned before, they are hidden in the sensing units in each group.

The sensing unit is shown in Fig. 7. Like all other levels of cache on this processor, the level-three cache uses single-ended sensing. During a read operation, a falling bitline selected by the local column select pulls down “DATA0” or “DATA1,” which pulls up the pre-discharged node “NOUT,” turning on a large NFET to pull down the precharged output of the sensing unit. The major advantage to this circuit is that it is fast and very small. However, because this is essentially a zipper-circuit,

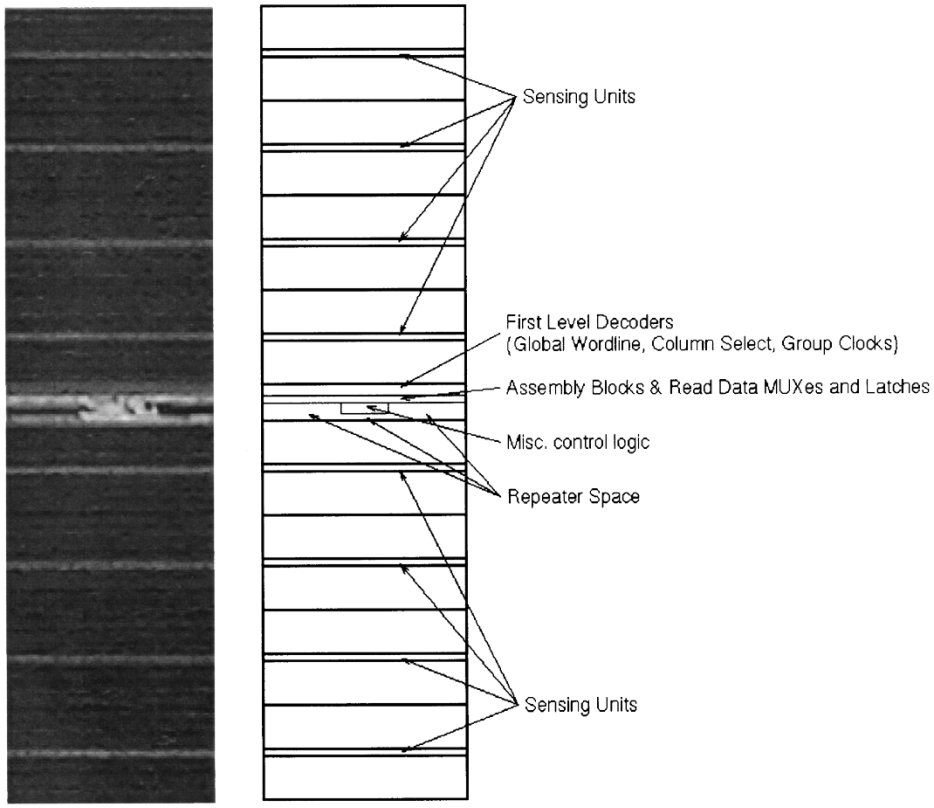


Fig. 5. Subarray layout.

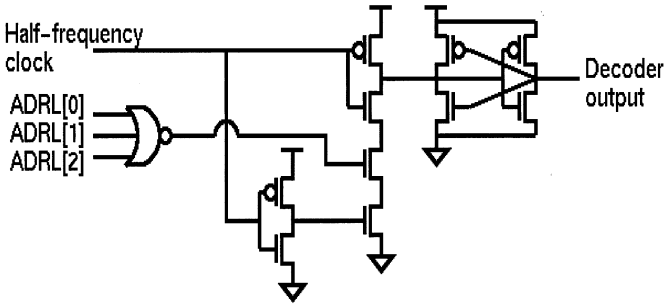


Fig. 6. First-level decoder.

holders were added and special care was taken to ensure the circuit's robustness. The output of this sensing unit is latched in the middle of the subarray, and is piped back to the core in four consecutive cycles, with the critical chunk first.

The write operation is very much like the read operation, where the full cache line is sent to the subarrays in four consecutive cycles before the write of the entire line can occur. The circuit illustrated in Fig. 8 is the write assembly circuit used for latching the write data, as well as to drive one-hot write data to the sensing units. In order to allow for a write to occur every four cycles, the circuit must allow for new data to be latched without disturbing a write in progress. The first stage of the circuit is the latch, while the second part is the driver. When the "Write" signal rises on rising clock, one of the two precharged nodes will discharge, depending on the data stored by the latch, firing either "Write1" or "Write0." As soon as one of these signals fire, the feedback circuit disables the pull-down chain, allowing new data to be latched. Race is prevented by allowing the latch to only be transparent during the second phase of the clock.

The one-hot "Write1" and "Write0" signals from the write assembly circuit go to the sensing unit, shown in Fig. 7, to enable dual-ended write into the SRAM cell. During a write, either "Write1" or "Write0" enables pulling down the selected bitline, while holders hold the bitline compliment high. The circuit here also supports weak write test mode (WWTM) functionality, which during manufacturing testing aids in detecting impedance faults in memory cells without data retention testing [6]. WWTM attempts to weakly overwrite a SRAM cell. Healthy cells will be able to retain their values, while defective cells will not. WWTM is usually implemented by having special WWTM cells attached to bitlines at both sides of the sensing circuit, and usually costs about 2%–3% in array efficiency. In the sensing circuit here, WWTM is implemented with essentially no area penalty. During WWTM, the "WWTML" NFET is simply turned off, allowing the carefully sized parallel NFET to weakly write into the SRAM cell. The penalty associated with this area efficient scheme is that, unlike the typical design, it cannot write to all eight bitlines of a sensing unit in parallel, costing $8\times$ in test time. However, because the write data can be deterministic during WWTM, care was taken to write either all 1's or all 0's during WWTM. This allows all the groups and the sensing units to be enabled at once, providing $32\times$ improvement in test time efficiency, which more than compensates for the $8\times$ penalty. The WWTM test of the entire array requires only 3032 cycles (96 rows *8 columns *4 passes).

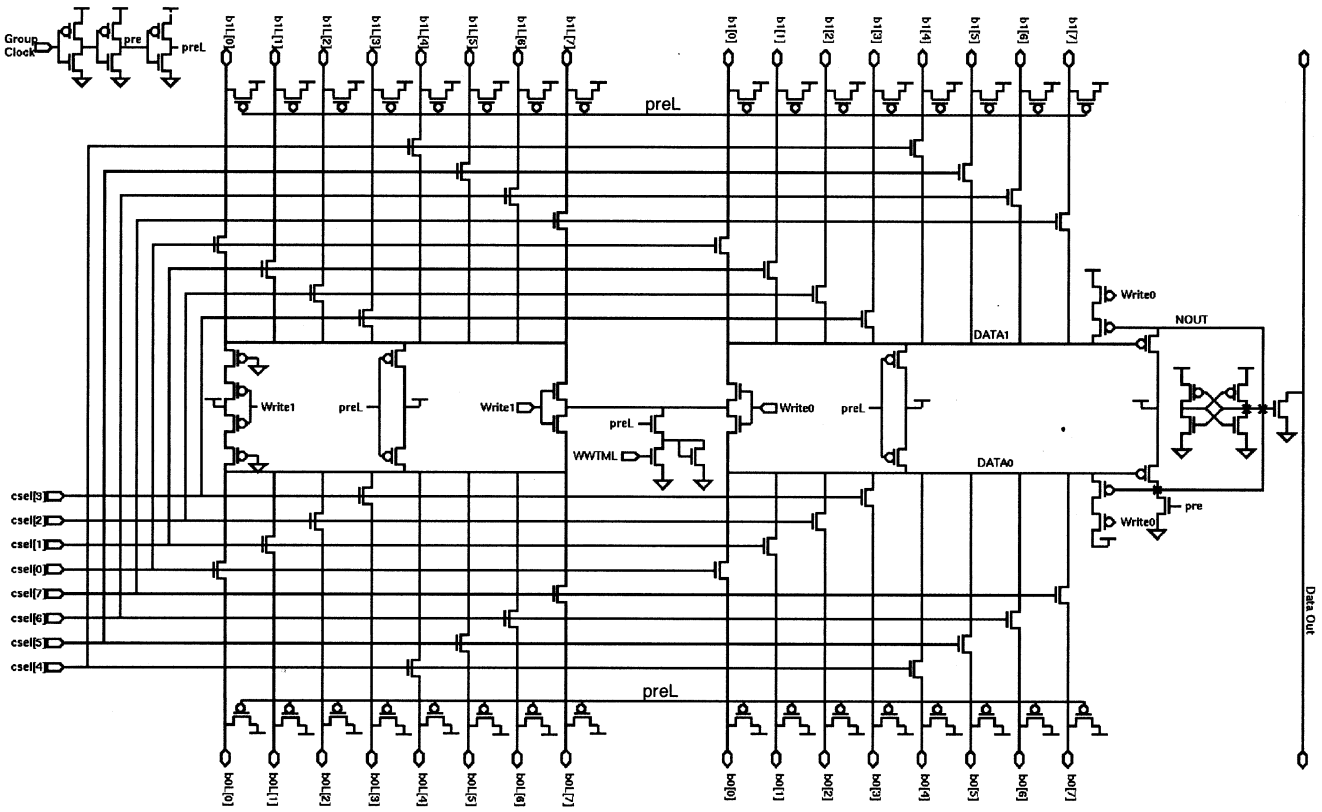


Fig. 7. Sensing unit.

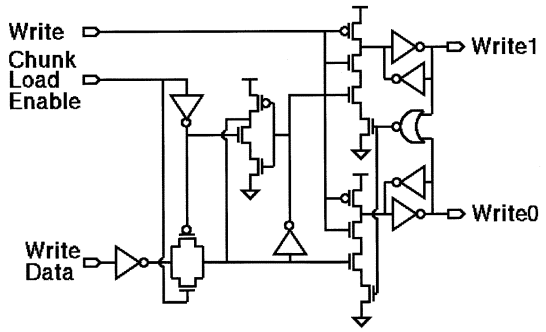


Fig. 8. Write assembly circuit.

VI. TIMING

The spice simulation in Fig. 9 shows the timing of a read operation. An access into the data arrays takes four cycles, from when the address is sent from the tag unit to when the first data is returned to the bus cluster unit. The first cycle is used mainly for address distribution from the tag unit to all the subarrays. It takes almost a full cycle because the longest path runs about 22 000 μm. Also, the address needs to reach every subarray, which creates very high fan-outs. Routing these high fan-out address and control signals is a challenge for the subarray design style. These signals are hand routed with wide metals and repeated with skewed inverters. Once the address reaches the subarrays, then the first-level decoding is performed. The second and third cycles are used for the actual array access. The rising clock edge fires the group clock, which eventually causes the local wordline to fire. The bitline falls and triggers the output of the sensing unit. The output gets latched by the second phase of the third cycle and does not get precharged until the next read or write operation. After going through the muxes, the data is returned during the fourth cycle to the bus cluster unit, where it goes through redundancy muxing. The data lines are autorouted to and from the subarrays over other subarrays and are repeated as necessary. Repeaters are placed in the global repeater space allocated in the middle of each subarray, as described in Section IV. Because data are piped back to the core in four cycles, two read accesses can occur no closer than four cycles apart. However, reads and writes can be interleaved together, which is shown in the simulation. So the cache can support one read and one write every four cycles.

Fig. 10 shows how those four read operation cycles fit in with the overall picture. The pipeline diagram contains the L3 pipeline and includes parts of the main pipeline and the L2 pipeline [1], [4]. The L3 cache's official load-use latency is 12 cycles. In the L3 pipeline, the first cycle is used for sending the request from the L2 to the tag unit. The next cycle is used for address distribution within the tag, as well as decoding. The tag access happens during the L3B cycle, and ECC correction occurs during its second phase. The L3C cycle is used for hit compare. The four cycles from L3D to L3G are the four cycles described before. Finally, the last cycle is used for sending the return data from the bus cluster to the level-two cache.

In order to reduce the number of clock buffers required to drive such a large area, the cache allows for 50 ps of clock skew in addition to the standard clock skew specified for the

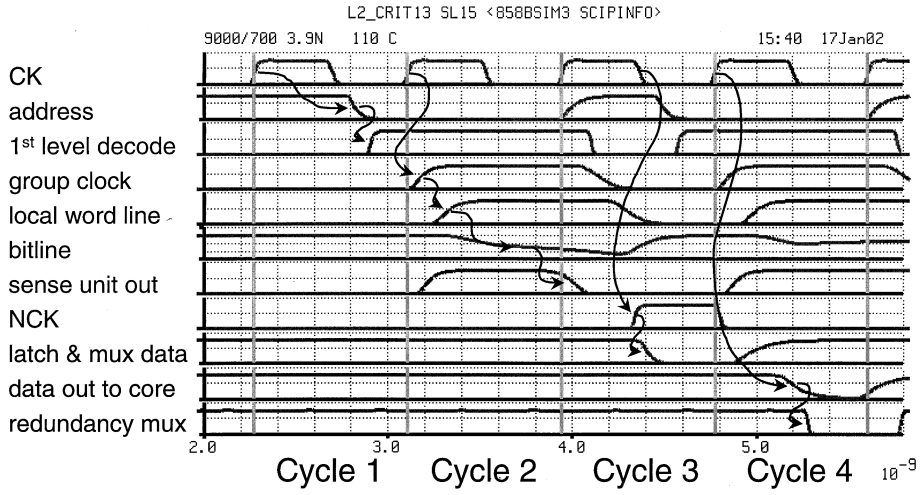
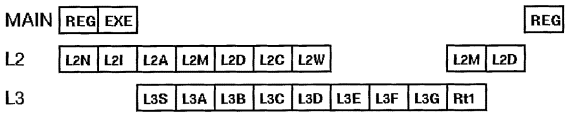


Fig. 9. Read operation timing.



L3S	Request issued from L2 to L3	L3D-L3G	L3 data array access
L3A-L3C	L3 tag access and hit compare	Rt1	Critical chunk return to L2 from BC

Fig. 10. Level three cache pipeline.

core blocks. The additional skew is taken into account during the design to satisfy both frequency and race requirements. Silicon testing showed that more than 105 ps of additional clock skew can be tolerated between 1.2 and 1.9 V before the cache experiences failures due to race.

VII. Test and Reliability Features

The cache's ECC detects double-bit errors and corrects single-bit errors. As mentioned before, the redundant subarrays can repair not only row and column defects, but whole subarray failures as well. A simplified BIST engine generates a March C- test pattern suitable for burn-in toggle coverage and power-on self-test. The cache also supports direct access test (DAT), which is direct parallel access into the cache through the I/O. During manufacturing debug, dedicated test patterns are applied to achieve high fault coverage and analyze specific failures. Finally, all control signals are scannable through JTAG.

VIII. Summary

We have presented a 3-MB level-three cache on the Itanium 2 processor. It uses a subarray design style, which efficiently utilizes available area and flexibly adapts to floor plan changes. Through distributed decoding and compact cell design and layout, 85% array efficiency was achieved. The cache can perform one read and one write every four cycles, providing a total bandwidth of 64 GB/s when running at 1 GHz. Finally, the cache has been characterized to operate above 1.2 GHz at 1.5 V and 110 °C.

Acknowledgment

The authors would like to thank S. Naffziger for technical contributions; J. Butler, J. Ignowski, R. Woodruff, L. Mamileti, S. Chakraborty, S. Stevens, T. Jackson, C. Tong, and J. Johnson for support.

References

[1] S. Naffziger *et al.*, "The implementation of the next-generation 64b Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, vol. 45, Feb. 2002, pp. 344–345.

[2] S. Rusu *et al.*, "The first IA-64 microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1539–1544, Nov. 2000.

[3] D. Bradley *et al.*, "The 16 kB single-cycle read access cache on a next-generation 64b Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, vol. 45, Feb. 2002, pp. 110–111.

[4] R. Riedlinger *et al.*, "The high-bandwidth 256 kB 2nd level cache on an Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, vol. 45, Feb. 2002, pp. 418–419.

[5] A. Ohba *et al.*, "A 7-ns 1-Mb BiCMOS ECL SRAM with shift redundancy," *IEEE J. Solid-State Circuits*, vol. 26, pp. 507–512, Apr. 1991.

[6] B. Bateman *et al.*, "A 450 MHz 512 kB second-level cache with a 3.6 GB/s data bandwidth," in *ISSCC Dig. Tech. Papers*, vol. 41, Feb. 1998, pp. 358–359.

Don Weiss received the B.S. and M.S. degrees in electrical and computer engineering from the University of Wisconsin-Madison in 1975 and 1976, respectively.

He joined Hewlett-Packard Company (HP), Loveland/Fort Collins, CO, in 1976 to work on HP's first 32-b microprocessor design. Since then, he has worked in test methodology, architecture verification, and VLSI circuit design on several PA-RISC microprocessor and I/O chips. His current main interest is embedded SRAM design.

John J. Wuu received the B.S. degree in electrical engineering and the M.Eng. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1997. His M.Eng. thesis involved the design of scaleable embedded SRAMs with Intel Corporation.

He joined the Hewlett-Packard Company, Fort Collins, CO, in 1997. Since 2000, he has also been teaching the "Introduction to Circuits and VLSI Design" class at the Fort Collins High School. The one-semester class seeks to teach VLSI design concepts to advanced high school students, allowing top students to work as part-time mask designers during the school year. His current interests include embedded SRAM design, high-speed circuit design, and teaching.

Victor Chin received the B.S. and M.E. degrees from the Massachusetts Institute of Technology, Cambridge, in 1994 and 1995, respectively.

After graduation, he joined Intel and was a key designer for the Pentium chipset, 430TX. From 1997 to 2002, he was a Senior Sesigner on Itanium2, code named "McKinley." He contributed to design of the high-performance units, FPU, and L3 tag cache. He is currently working on future Itanium designs.